## REMARKS

For the record, the Office Action, page 2 item 1 indicated that the action is in response to Applicants' response dated 12/11/2009. This appears to be a typographical error, as the most recent response in the file of this application according to PAIR is dated 2/12/2009. Accordingly, Applicants' amendments and remarks are based upon the claims and response dated February 11, 2009.

Examiner Adnan Mirza and Lilian Y. Ficht, representative for the Applicants, held a telephonic interview on August 6, 2009. At this interview, proposed claim amendments to claim 1 were presented. Applicants also explained the material of the present application, and discussed differences between the present application and the cited art. Applicants thank the Examiner for his time and consideration.

Claims 5-6 are canceled. The material of canceled claims 5-6 is incorporated into claims 1 and 4. Claims 1-4 and 17-27 are pending in this application.

Claims 1, 9 and 17 are amended. Support for the amendments to claims 1, 9 and 17 may be found, for example, in the specification at paragraphs [0001]-[0002], [0005]-[0006], -0009], [0025]-[0028], [0034], [0036]-[0037], [0039], [0041], [0117] and in Figure 3. Claims 25-27 are similarly amended.

Claim 2 is amended. Support for the amendments to claim 2 may be found, for example, in the specification at paragraph [0042]. Claim 3 is amended. Support for the amendments to claim 3 may be found, for example, in the specification at paragraph [0037]. Claim 4 is amended. Support for the amendments to claim 4 may be found, for example, in the specification at paragraph [0037].

Claims 7 and 8 are amended to reflect changes made to claim 1. Claims 10 and 12-14 are amended to reflect changes made to claim 9. Claims 18 and 20-24 are amended to reflect changes made to claim 17. Claim 21 is further amended to correct antecedent basis.

Thus, no new matter is added with these amendments.

Claim Rejections Under 35 USC §103

Applicants traverse the rejection of claims 1-4 and 7-27 under 35 U.S.C. 103(a) as being unpatentable over *Bian et al.*, U.S. Patent No. 7,509,656 ("Bian") and further in view of *Dongarra et al.*, University of Tennessee-Knoxville "Using PAPI for Hardware Performance Monitoring on Linux Cluster" ("Dongarra"). Applicants respectfully request reconsideration and withdrawal of the rejection in light of the amendments made to the claims.

Bian is generally directed to providing an application program interface (API) to abstract a count history from a hardware counter up to software so that the hardware count history is accessible to an application program (Bian, 2:34-38). The abstracted hardware counts are located within the memory space of the API (Bian, Fig. 1). Dongarra is generally directed to providing a platform-independent API interface (referred to by Dongarra as "Performance API" or "PAPI") to access hardware performance counts. (Dongarra, abstract). Dongarra's APIs may access predefined counts across different hardware platforms as well as provides access to hardware count events specific to a particular platform (Dongarra, sections 4.1 and 4.2). Thus, both Bian and Dongarra address hardware performance counting and API interfaces thereto.

The present application, however, is concerned with software performance counts, e.g., performance of operating system components, queue length, thread CPU time, database cache size, etc. (specification, [0001], [0004]-[0005]). For each instance of a software performance counter provider application process, a separate stored data subset may be uniquely identified by the GUID of the corresponding performance counter provider application process (specification, [0037]-[0038]).

First, the systems of Bian and Dongarra collect and provide hardware performance counts, but do not disclose providing performance counts associated with software performance. The APIs disclosed in Bian and Dongarra are called by performance counter consumers, not performance counter providers.

Secondly, the systems of Bian and Dongarra, are not able to store or distinguish sets of stored counts by software performance provider application process

instances, or, for that matter by any provider instances. This drawback may result in potential inaccuracies, and performance counts may not be directly attributed to the correct, corresponding count providers. For example, Bian resets hardware counters when they are saturated without any disclosed regard to what processes or instances may have been executing at the time. Data may be lost, muddled and not attributable to a certain instance of an application. Bian, 7:10-13 admits this problem: "That is, there is no way of knowing how much data was lost from the time the hardware counter(s) saturated, rendering the count value(s) in the software counter(s) unreliable." Dongarra also admits difficulties with counter accuracy: "Problems with the accurate attribution of hardware events to individual instructions in out-of-order processors is a well-known problem... We plan to collaborate with other researchers ... for determining the accuracy of hardware counter interfaces" (Dongarra, Section 7). Neither Bian nor Dongarra provides a unique reference to a specific counter data set for an instance of a performance counter provider application process, so that performance counts in the specific counter data set may be accurately maintained and attributed to the correct instance of a performance counter provider application.

These differences and advantages of the present application over Bian and Dongarra are detailed in the discussion of claim elements below.

**Claim 1:**

Applicants respectfully submit that a *prima facie* case of obviousness cannot be established for claim 1, as amended, by Bian and Dongarra, as no combination of Bian and Dongarra teaches, suggests or discloses each and every element of claim 1, as amended.

Performance Counter Provider Application Process and Software Performance Counts:

For example, amended claim 1 recites: "a performance counter provider application process executed by the processor, the performance counter provider application process enabled to generate raw performance counter information corresponding to software performance in the computer system." Neither Bian nor Dongarra discloses this element, as both references teach hardware counters generating hardware performance data but not application processes generating software performance data. Bian does disclose an application program 130, but the application program 130 does not *generate* raw performance counter information corresponding to software performance. Instead, the application

program 130 *receives* and *uses* hardware performance data: "One of the functions performed by application program 130 relies on count data from hardware counter 120" (Bian, 2:62-64). Thus, Bian's application program 130 is a performance counter consumer, not a performance counter provider. Similarly, Bian's application programs 340 are also performance counter consumers: "One or more applications 340 in the system memory 330 may use count data from various ones of the hardware counters..." (Bian, 3:64-67). Thus, Bian only teaches performance counter consumer application programs, and not the "performance counter provider application process" recited in claim 1.

Similarly, Dongarra is directed to APIs that provide counts from hardware counters to applications (Dongarra, abstract). Dongarra's counts relate to hardware performance, are generated by hardware counters, and are provided to performance count consumers via Dongarra's APIs. Accordingly, Dongarra does not teach software performance counters at all, let alone "a performance counter provider application process... enabled to generate raw performance counter information corresponding to software performance in the computer system." Therefore, Dongarra also does not teach, suggest or disclose this missing element.

Counter Provider API:

Furthermore, Applicants submit that neither Bian nor Dongarra teaches, suggests or discloses the element of amended claim 1:

> a counter provider application program interface (API) of the operating system, *called by the performance counter provider application process* and enabled to... (italics added)

The APIs taught in Bian and Dongarra are invoked by performance counter *consumer* processes, not performance counter *provider* processes, as called for by claim 1. For example, as previously discussed, Bian's application program 130 is a counter consumer. Bian teaches his API as being invoked by his counter consumer application program 130: "application program 130 issues a command, or function call, to API 140 to instruct API 140 to access the hardware counter" (Bian: 3, 1-3). Dongarra teaches both his high-level and low-level interfaces being invoked to retrieve counts for count consumers: "PAPI provides two interfaces to the underlying counter hardware: a simple, high level interface for the acquisition of simple measurements, and a fully programmable, thread safe, low level

interface... [that] provides access to all available native events and counting modes" (Dongarra, Introduction, paragraph 2). Thus, for at least this reason, neither Bian nor Dongarra teaches, suggests or discloses this missing element.

Additionally, with further regard to the missing counter provider API element, the missing element further recites:

> provide a counter query function called by the performance counter consumer application process and enabled to:
>
> extract the GUID of the performance counter provider application process from the repository, and
>
> retrieve counter data corresponding to the GUID of the performance counter provider application process from the performance counter structure within the address space of the performance counter provider application process by invoking the access function via the operating system.

Applicants respectfully submit that neither Bian nor Dongarra teaches, suggests or discloses this missing element. As previously discussed, neither Bian nor Dongarra describes a performance counter provider application process, so neither Bian nor Dongarra can teach, suggest or disclose extracting a GUID of this missing application process, let alone a counter query function provided by the same API called by the performance counter provider application process enabled to extract the GUID from a repository and retrieve counter data corresponding to the GUID.

Moreover, with further regard to this missing element, Applicants respectfully submit that neither Bian nor Dongarra teaches, suggests or discloses any of their APIs as performing "register[ing] a description...," "allocat[ing] a performance counter structure within an address space...," "automatically assign[ing] an access function... enabled to be invoked via the operating system...," or "during a run-time of the counter provider application process, register[ing] a performance counter provider dataset..." This makes sense, as both Bian and Dongarra describe APIs used by the consumer side of hardware performance counting, whereas the counter provider API of the claim supports the provider side of software performance counting.

Therefore, no combination of Bian and Dongarra teaches, suggests or discloses each and every element of claim 1, as amended. For at least this reason, Applicants respectfully submit a *prima facie* case of obviousness cannot be established for claim 1 by Bian and Dongarra. Accordingly, Applicants respectfully request withdrawal of the rejection.

**Claims 9, 17, 25-27:**

Independent claims 9, 17, 25-27 are each amended similarly to claim 1. For at least reasons similar to amended independent claim 1, Applicants respectfully submit no combination of Bian and Dongarra teaches, suggests or discloses each and every element of independent claims 9, 17, 25-27, as amended. For at least this reason, Applicants respectfully submit a *prima facie* case of obviousness cannot be established for independent claims 9, 17, 25-27 by Bian and Dongarra. Accordingly, Applicants respectfully request withdrawal of the rejection.

**Claims 2-4, 7-8, 10-16 and 18-24:**

Claims 2-4 and 7-8 each depend from amended independent claim 1. Claims 10-16 each depend from amended independent claim 9. Claims 18-24 each depend from amended independent claim 17. Independent claims 1, 9 and 17 have been demonstrated to be allowable over Bian and Dongarra. Dependent claims incorporate by reference each and every element of their respective independent claim and any intervening claims. For at least this reason, Applicants respectfully submit no combination of Bian and Dongarra teaches, suggests or discloses each and every element of dependent claims 2-4, 7-8, 10-16 and 18-24.

Accordingly, Applicants respectfully submit a *prima facie* case of obviousness cannot be established for dependent claims 2-4, 7-8, 10-16 and 18-24 by Bian and Dongarra, and Applicants respectfully request withdrawal of the rejection.

**CONCLUSION**

In view of the above amendment and arguments, the Applicants submit the pending application is in condition for allowance and an early action so indicating is respectfully requested.

The Commissioner is authorized to charge any fee deficiency required by this paper, or credit any overpayment, to Deposit Account No. 13-2855, under Order No. 30835/302623, from which the undersigned is authorized to draw.

Dated: August 13, 2009

Respectfully submitted,

By___/Lilian Y. Ficht #64,514/_____
Lilian Y. Ficht
Registration No.: 64,514
MARSHALL, GERSTEIN & BORUN LLP
233 S. Wacker Drive, Suite 6300
Sears Tower
Chicago, Illinois 60606-6357
(312) 474-6300
Patent Agent for Applicants